CLAIMS

What is claimed is:

1. A method comprising:

predicting a next micro-operation address;

storing the predicted address into a first memory;

retrieving the predicted address from the first memory;

accessing a second memory at the retreived address to get a next micro-operation.

2. The method of claim 1, wherein storing the predicted address comprises programming the address into a read-only memory.

3. The method of claim 1, further comprising determining whether the micro-operation address is correctly predicted.

4. The method of claim 3, further comprising correcting the predicted address if the address is mispredicted.

5. The method of claim 4, wherein the next micro-operation indicates whether there is a jump present.

6. The method of claim 5, wherein the next micro-operation comprises one or more jump bits.

7. The method of claim 6, wherein determining whether the address is correctly predicted comprises checking the jump bit of the next micro-operation.

8. The method of claim 7, wherein the next micro-operation address comprises a plurality of bits.

9. The method of claim 8, wherein determining whether the address is correctly predicted further comprises checking the two least significant bits of the next micro-operation address to determine if a jump was executed.

10. The method of claim 9, wherein correcting the predicted address comprises zeroing out the two least significant bits of the next micro-operation address.

11. The method of claim 1, further comprising storing the next micro-operation for use in an instruction pipeline.

12. The method of claim 11, wherein storing the next micro-operation comprises writing the micro-operation into a register.

13. A system comprising:

a first memory to store microcode, wherein the first memory is accessed at a next address to get a next micro-operation;

a second memory to store predicted micro-operation addresses;

misprediction recovery logic coupled to the first memory to determine if the predicted address is correct and to determine a recovery address; and

a selector coupled to the first memory, the second memory, and the misprediction recovery, to select either the predicted address or the recovery address

as the next address at which to access the first memory based on the determination by the misprediction recovery logic as to whether the predicted address is correct.

14. The system of claim 13, wherein the misprediction recovery logic to determine if the predicted address is correct comprises the misprediction recovery logic to determine whether there is a jump present and whether a jump was executed.

15. The system of claim 14, wherein each address comprises a plurality of bits.

16. The system of claim 15, wherein the next micro-operation comprises at least one jump bit.

17. The system of claim 16, wherein the misprediction recovery logic to determine whether there was a jump present comprises the misprediction recovery logic to check the jump bit of each micro-operation.

18. The system of claim 17, wherein the misprediction recovery logic to determine whether there was a jump executed comprises the misprediction recovery logic to check the two least significant bits of the next address.

19. The system of claim 18, wherein the misprediction recovery logic to determine the recovery address comprises the misprediction recovery logic to zero out the two least significant bits of the next address.

20. The system of claim 19, wherein the misprediction recovery logic to determine the recovery address further comprises the misprediction recovery logic to add the number of micro-operations per line to the next address.

21. The system of claim 13, further comprising a register coupled to the first memory to store the next micro-operation.

22. The system of claim 13, further comprising a register coupled to the first memory to store the next address for use by the misprediction recovery logic.

23. The system of claim 13, wherein the selector is a multiplexer.

24. A method comprising:

predicting a next micro-operation address;

determining a recovery address;

determining whether the predicted address is correct;

selecting between the predicted address and the recovery address based on whether the predicted address is correct; and

accessing a memory with the selected address to get the next micro-operation.

25. The method of claim 24, further comprising storing the predicted address.

26. The method of claim 25, wherein storing the predicted address comprises storing the predicted address in a read-only memory.

27. The method of claim 24, wherein determining whether the predicted address is correct comprises determining whether there is a jump present and whether a jump was executed.

28. The method of claim 24, wherein the memory stores microcode.

1 29. The method of claim 24, further comprising storing the next micro-operation.

1

1 30. The method of claim 24, further comprising storing the selected address.